

Problema 2

Implementação

qr_GSM.m

```
% Entradas:
% A - matriz (m x n)
% Saídas:
% Q = matriz (m x n) ortogonal
% R = matriz (n x n) triangular superior
function [Q,R] = qr_GSM(A)
    [m, n] = size(A);

    % Inicializa matrizes
    Q = zeros(m,n);
    R = zeros(n);

    for j = 1 : n
        v = A(:,j); % j-ésima coluna de A

        % Obtém, por Gram-Schmidt, v o j-ésimo vetor de uma base ortogonal
        for i = 1 : j-1
            R(i,j) = dot(Q(:,i), v); % Usa o vetor atualizado
            v = v - R(i,j) * Q(:,i);
        end

        R(j,j) = norm(v,2);
        Q(:,j) = v / R(j,j); % j-ésimo vetor de uma base ortonormal
    end
end
```

Testes

Para a matriz A, essa modificação não surtiu efeito algum. Portanto, não há necessidade expor os resultados.

Para a matriz B, uma matriz má condicionada, a modificação alterou minimamente uma entrada da matriz R. Quanto à matriz Q, notamos que a ultima colunas dessa nova Q é diferente da anterior.

```
>> QCb(:,4)' % Quarta coluna da matriz Q obtida com qr_GSC
    0.32233    0.40291    0.64466   -0.56408

>> [QMb, RMb] = qr_GSM(B);

>> QMb(:,4)'
    0.94679    0.063119    0.25248   -0.18936
```